



**DISTRIBUTED MEANS OF ORGANIZING AN
ARBITRARILY LARGE NUMBER OF COMPUTERS**

BACKGROUND OF THE INVENTION

Technical Field of the Invention

This invention relates to the fields of computer
5 communications and computer networking.

Description of Related Art

The invention describes a means of coordinating a
plurality of computers physically connected to a network.
The invention allows rapid broadcast and search of the
10 hosts using distributed control.

Such a system is an improvement over existing
centrally controlled coordination in the areas of

security, attack resistance, robustness, speed, and scalability.

Physical vs. Logical Organization

5 The invention may be used by any computer possessing a means to communicate one to another with all other computers to be coordinated.

10 The invention works by arranging the physically connected computers into a virtual hierarchy that may be efficiently searched and to which messages may be broadcast.

Prior Art - Centralized Control

15 A collection of computers possessing the ability to communicate one to another may be coordinated by a controlling computer. The controlling computer can maintain a list of network addresses of the computer collection. The controlling computer can maintain a list describing the contents available on each computer in the collection. The controlling computer can also maintain a list a names or other identifying characteristics of each
20 computer in the collection.

5 The controlling computer may broadcast a message to all the coordinated computers by reading its list of network addresses and creating and sending a message to each computer in the list. The controlling computer may respond to a query regarding the contents available in the collection of computers by searching its list of contents. When a content match is found, the controlling computer can provide the network address of the computer possessing the content.

10 The controlling computer may respond to a query regarding the name or other identifying characteristic of a computer in the collection by searching its list of names or other identifying characteristics of each computer in the collection. When a match to the computer
15 name or other identifying characteristic is found, the controlling computer can provide the network address of the computer in the collection matching the desired name or other identifying characteristic.

Disadvantages of Centralized Control

- 20
1. Scalability
 2. Security
 3. Failure sensitivity

Scalability

A controlling computer can coordinate a finite number of computers. The amount of computing power increases linearly with the number of computers to be coordinated.

5 At some point, the controlling computer will be unable to coordinate any more computers and the collection of computers must cease increasing.

Security

10 All communications between the collection of computer is coordinated by the controlling computer. A successful security attack on the controlling computer will disable all coordinated communications with the collection of computers.

Failure Sensitivity

15 Similarly, a software failure or hardware failure on the single controlling computer can disable coordinated communications with the entire collection of computers.

Prior Art - Hierarchical Organizations

20 A logical communication hierarchy of computers may be created with a single host at the top of the logical

hierarchy and two or more computers at the next level.
Each one of the computers at the second level is logically
connected to two or more computers. Each of the computers
at the third level is logically connected to two or more
5 computers, and so on.

In the simplest case each computer is connected to a
single host going up the hierarchy and to two computers
going down the hierarchy. The top level of the hierarchy
has a single computer. The second level has two
10 computers. The third level has four computers. The
fourth level has eight computers. Each successive logical
level contains the next power of 2 number of computers.

The total number of computers in the hierarchy is the
sum of the computers at each of the N levels in the
15 hierarchy. For example in the simple sample just
described, the fifth level would contain 32 computers and
the entire hierarchy would contain 63 computers.

The tenth level of a ten level hierarchy would
contain 1024 computers and the entire hierarchy would
20 contain 2047 computers. The equation describing the total
number of computers in such a network is $2 * (2^{\text{power } N}) - 1$.
Where N is the number of levels.

Advantage of Hierarchical Organizations

A virtual hierarchy such as that described above may broadcast a message from the top level computer to all computers in the hierarchy is an amount of time equivalent
5 to:

$N * (\text{average message transit time}).$

For example, in an Internet environment, the message delay might be 10 msec. To broadcast a message to all computers in a ten level hierarchy containing 2047 computers would
10 take 100 msec.

To broadcast a message to all computers in a twenty level hierarchy containing 2,097,151 computers would take only 200 msec.

Without a hierarchy, serially sending a message to 2
15 million computers would require $2,097,151 * (\text{average message transit time}).$

Disadvantages of Hierarchical Organization

1. Inflexibility
2. Insecurity
- 20 3. Failure sensitivity

Inflexibility

A hierarchy is great for broadcasting a message from a constant fixed computer to a hierarchy of constant fixed computers. Should a message require broadcasting from any other computer than that at the top of the hierarchy, other mechanisms must be used such as passing the message upward to the top and then broadcasting it downward from the top of the hierarchy.

Insecurity

A hierarchy is progressively more vulnerable to disruption the closer to the top of the hierarchy one gets. In fact, if an attack were launched on the top computer of the hierarchy rendering it inoperable, the entire hierarchy would be rendered inoperable.

Failure Sensitivity

Similarly, each computer in a hierarchy is connected to two or more computers beneath it. Each of them is connected to two or more computers and so on. Should a single computer in a hierarchy fail, all the computer connected beneath it in the hierarchy would be unable to communicate.

SUMMARY OF THE INVENTION

Some of the objectives of the invention are to
provide a collection of computers with:

- (a) the ability to perform one to many broadcast
5 originating from any computer in the collection without
requiring a fixed hierarchical organization;
- (b) the ability to search all computers in the collection
for specific content without requiring a central
controlling computer;
- 10 (c) the ability to locate a specific computer in the
collection using a name or other identifying
characteristic without requiring a central controlling
computer;
- (d) the ability to self organize itself so broadcast and
15 searching may be performed in an optimal manner;
- (e) the ability to self organize itself in the presence of
changing computer network addresses so broadcast and
searching may be performed in an optimal manner;
- (f) the ability to continue broadcast and searching
20 functions in the presence of a security attack which
disables a plurality of computers in the collection.

(g) the ability to repair itself and continue broadcast and searching functions in the presence of simultaneous failure of a plurality of computers in the collection.

Further objectives will become apparent from a
5 consideration of the ensuing description and drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention, reference is made to the following detailed description taken in conjunction with the accompanying
10 drawings wherein:

Fig 1A to 1B show a typical physical means of computers communicating one to another compared to a logical means of communicating for the same computers.

Fig 2 shows a typical embodiment of the invention
15 using a quadrilateral configuration for the collection of computers.

Fig 3A and 3B show a computer's neighbors and the linking of neighbors in the collection of computers.

Fig 4 shows a typical embodiment of the invention
20 using a hexagonal configuration for the collection of computers.

Figs 5A to 5C show the virtual hierarchy created by a typical embodiment of the invention using a quadrilateral configuration for the collection of computers.

5 Figs 6A to 6C show the path of a broadcast message originating from the top computer of the hierarchy.

Fig 7A and 7B show a new computer being added to the collection of computers.

10 Fig 8A and 8B shows a non-optimized view of a quadrilateral configuration for the collection of computers and an optimized view of the same collection of computers.

Figs 9A-1 to 9P show the self organizing states of a quadrilateral configuration for the collection of computers.

15 Figs 10A to 10D show a view of a quadrilateral configuration with three missing computers.

Fig 11 shows the organization of computers with a subset of computers performing caching.

Reference Numerals in Drawings

20 10 logical middle of the collection of computers
11 top computer of a hierarchy
12,13,14,15,16,17,18,19,20,21,22,23 computers

- 24 physical means of communication 25 logical means
of communication
- 26 radial 27 indirect radial
- 28 found computer 29 disabled computer
- 5 30 collection edge 31 logical connection
- 32 new computer 33 caching computer
- 34 searching computer
- 40 head 41 right hand
- 42 toe 43 left hand
- 10 44 WEBcast server 45 WEBcast player

DETAILED DESCRIPTION OF THE INVENTION

Reference is now made to the Drawings wherein like reference characters denote like or similar parts throughout the various Figures.

15 Description-Figs. 1 to 3

Each computer 10,12,13,14,15 in the collection possesses a physical means 24 to communicate one to another as illustrated in Fig 1A. The physical means of communication may be reorganized to produce a logical
20 means to communicate 25 one to another as illustrated in Fig 1B.

In an Internet embodiment, Fig 1A would consist of computers physically connected via local area networks, routers, and the Internet backbone. In an Internet embodiment, Fig 1B would consist of computers logically
5 connected via (IP) Internet Protocol addresses.

The present invention is primarily directed to the logical configuration of a computer network, irrespective of the particular physical means of communication between computers in the network. A typical embodiment of the
10 invention in a quadrilateral configuration is illustrated in Fig 2. The logical middle 10 of the collection possesses a logical means to communicate 25 to four neighbor computers 12,13,14,15 that comprise a concentric square. Each computer in the collection possesses a
15 logical means to communicate 25 to four neighbors. The concentric square consisting of four computers 12,13,14,15 is in turn logically connected with the concentric square consisting of eight computers 16,17,18,19,20,21,22,23. The collection of computers ends at the collection edge
20 30.

The logical means to communicate 25 is defined by a neighbor relationship. Each computer possesses links to four neighbors. Fig 3A illustrates each computer's

neighbors as neighbor0, 40, neighbor1 41, neighbor2 42,
and neighbor3 43. Neighbor1 41 is to the right of (or
counterclockwise from) neighbor0 40, neighbor2 42 is to
the right of (or counterclockwise from) neighbor1 41, and
5 neighbor3 43 is to the right of (or counterclockwise from)
neighbor2 42.

In Fig 3B the neighbor0s 40 of four computers
12,13,14,15 point to the logical center 10 of the
collection. The neighbor2s 42 of these computers
10 12,13,14,15 point to four more computers
16,17,18,19,20,21,22,23 and so on.

The neighbor2 42 to neighbor0 40 configuration of the
computers emanating from the logical center 10 of the
collection may be considered radials 26. The radials 26
15 pass through the computers at the vertices of the
concentric quadrilaterals. The neighbor0s 40 of
computers on a radial point toward the logical middle of
the collection of computers. The neighbor2s 42 of
computers on a radial point toward the collection edge 30.

20 The neighbor 41 of every computer on a radial 26
point to a line of neighbor2 42 to neighbor0 40
relationships which form indirect radials 27. The
neighbor0s 40 of computers on an indirect radial 27 point

toward a radial 26. The neighbor2s 42 of computers on an indirect radial 27 point toward the collection edge 30. Generally, the neighbor relationships are used for logically organizing the computers, while the radials 26 and indirect radials are used for disseminating information throughout the collection of computers.

A typical embodiment of the invention in a hexagonal configuration is illustrated in Fig 4. As will be appreciated by persons of ordinary skill in the art, the hexagonal configuration possesses six radials 26 emanating from the logical center 10 of the collection, while indirect radials 27 generally emanate from a neighbor 2 (out of a total of six neighbors) of every computer on a radial 26. Although the invention is described in connection with a quadrilateral or hexagonal configuration, configurations with any even plurality of sides are possible in accordance with the invention. Three dimension configurations are also possible. The more sides in the configuration, the more states exist in managing the collection of computers and the faster the broadcast and search of the entire collection may be performed. In any event, it will be recognized that,

among other things, the orientation of the indirect radial depends on the particular configuration.

Operation-Figs 5 to 7

5 The invention creates a logical organization in which each and every computer in the collection appears to be at the top of a hierarchy formed by concentric polygons. For purposes of describing the invention, the logical middle 10 of the collection of computers corresponds to the geometrical center, while the top computer 11 in the hierarchy represents a location at which a search or broadcast originates. As a result, the top computer 11, at any given time, changes based on which computer in the overall hierarchy is acting as a source of a broadcast, search, or other dissemination of information throughout 15 the network or a portion thereof. Fig 5A illustrates a hierarchy emanating from the top computer 11 in the hierarchy through a collection of computers 12,13,14,15,16,17,18,19,20,21,22,23 organized into concentric squares. In the example illustrated by Fig 5A, 20 the top computer 11 in the hierarchy is also the logical middle 10 of the collection of computers.

In the case of the quadrilateral configuration, the number of computers in the concentric squares increases by four from concentric square to concentric square. The first square surrounding the top computer 11 in the hierarchy contains four computers 12,13,14,15. The next concentric square contains eight computers 16,17,18,19,20,21,22,23. The next concentric square contains twelve computers, and so on.

Fig 5B illustrates a hierarchy emanating from the top computer 11 in the hierarchy through a collection of computers 12,13,14,15,16,17,18,19,20,21,22,23 organized into concentric squares. In the example illustrated by Fig 5B, the top computer 11 in the hierarchy is not the logical middle 10 of the collection of computers.

The number of computers in the concentric squares increases by four from concentric square to concentric square. The first square surrounding the top computer 11 in the hierarchy contains four computers 12,13,14,15. The next concentric square contains eight computers 16,17,18,19,20,21,22,23. The next concentric square (not fully shown) contains twelve computers, and so on.

Fig 5C illustrates a hierarchy emanating from the top computer 11 in the hierarchy through a collection of

computers 12,13,14,15,16,17,18,19,20,21,22,23 organized into concentric squares. In the example illustrated by Fig 5C, the top computer 11 in the hierarchy resides on the collection edge 30.

5 The number of computers in the concentric squares increases by four from concentric square to concentric square except computers residing on the collection edge 30 which do not possess a complete square of neighbors. In this case, therefore, the first square surrounding the top
10 computer 11 in the hierarchy contains two computers 12,13. The next concentric square contains five computers 16,17,18,19,20. The next concentric square (not fully shown) contains six computers. The next concentric square contains nine computers, and so on.

15 Message Broadcast

 One-to-many message broadcast may efficiently be performed by the collection of computers as each computer passes the broadcast to either one or two more computers in the next concentric polygon. Fig 6 illustrates the
20 rapid increase in number of computers reached with the addition of each concentric polygon.

Another look at Fig 5A illustrates how a broadcast operation may use this configuration to relay a message from the top of the hierarchy out through the concentric squares of computers. The message is initially passed
5 from the top computer 11 in the hierarchy to the four computers in the surrounding concentric square 12,13,14,15. In the example illustrated by Fig 5A, the top computer 11 in the hierarchy is also the logical middle 10 of the collection of computers.

10 The message is then passed by the computers at the four corners of the square 12,13,14,15 to the eight computers in the surrounding concentric square 16,17,18,19,20,21,22,23. The message is then passed to the computers in the next concentric square and so on.

15 Another look at Fig 5B illustrates how a broadcast operation may use this configuration to relay a message from the top of the hierarchy out through the concentric squares of computers. The message is initially passed from the top computer 11 in the hierarchy to the four
20 computers in the surrounding concentric square 12,13,14,15. In the example illustrated by Fig 5B, the top computer 11 in the hierarchy is not the logical middle 10 of the collection of computers.

The message is then passed by the computers at the four corners of the square 12,13,14,15 to the eight computers in the surrounding concentric square 16,17,18,19,20,21,22,23. The message is then passed to the computers in the next concentric square and so on.

Another look at Fig 5C illustrates how a broadcast operation may use this configuration to relay a message from the top of the hierarchy out through the concentric squares of computers. In the example illustrated by Fig 5C, the top computer 11 in the hierarchy resides on the collection edge 30. Computers residing on the collection edge 30 do not possess a complete square of neighbors.

The message is initially passed from the top computer 11 in the hierarchy to the two computers 12,13 in the surrounding but incomplete concentric square. The message is then passed to the five computers 12,13,14,15,16 in the surrounding but incomplete concentric square. The message is then passed to the computers in the next surrounding but incomplete concentric square and so on.

As can be seen in both Figures 5B and 5C, there is a possibility that one or more computers will be orphaned (e.g., computer 17 in Figure 5C), at least with respect to a message that does not originate from a top computer 11

at the logical middle 10 of the collection of computers.
The existence of such orphaned computers, however, does
not create a problem in most applications. Preferably,
although not necessarily, a broadcast will originate from
5 a top computer 11 that is at the logical middle of the
collection of computers. Thus, the broadcast will reach
all of the computers in the collection. On the other hand,
messages that originate from a top computer 11 that is not
at the logical middle 10 frequently do not need to reach
10 all of the computers in the collection. For example, a
search for a specified file or application normally does
not have to be exhaustive because there is often
sufficient redundancy in a computer network to locate a
subject of a search without querying every computer in the
15 network.

Searching the Collection of Computers for Content

A search of content residing on computers in the
collection may be performed if the message passed through
the hierarchy contains search parameters and the network
20 address of the requesting computer. The message will
radiate through the hierarchy causing each computer in the
collection to search its local content. When a computer

being searched finds a match to the searched content, that computer contacts the network address of the requesting computer and identifies the network address where the content is located.

5 Locating a Computer by a Name or Other Identifying Characteristic.

10 A computer in the collection of computers may be located by a name or other identifying characteristic if the message passed through the hierarchy contains the name or other identifying characteristic and the network address of the requesting computer. The message will radiate through the hierarchy causing each computer in the collection to compare the name or other identifying characteristic to its own name or other identifying
15 characteristic.

Initially Creating the Collection of Computers

The collection of computers must begin with a single computer which will be the initial logical middle 10 of the collection of computers. The next four additions to
20 the collection of computers will constitute the neighbor0

40, neighbor1 41, neighbor2 42, and neighbor3 43 neighbors
of the logical middle 10 of the collection of computers.

Adding Additional Computers to the Collection of Computers

In order to become part of the collection of
5 computers, a computer must first find another computer
which is part of the collection of computers.

A computer can find to the collection of computers
using several techniques including but not limited to:

(a) Trying to attach to a known list of available
10 computers in the collection of computers.

(b) Trying to attach to a computer from a list of
computers with which this computer has previously
communicated.

(c) Seeking a computer connected to the collection of
15 computers by broadcasting queries to network addresses
created from an algorithm such as a random number
generator.

When a new computer 32 locates a computer belonging
to the collection of computers, the found computer 28 (see
20 Figure 7A) will reside either on a radial 26 or indirect
radial 27. The neighbor2 42 of the found computer 28 will
point toward the collection edge 30. By following the

found computer's 28 neighbor's neighbor's 42 one after another, the collection edge 30 may be located. The new computer 32 is attached to the collection of computers at the edge 30.

5 Fig 7A illustrates following a radial 26 from the found computer 28 to the collection edge 30 to attach a new computer 32.

10 Fig 7B illustrates following an indirect radial 27 from the found computer 28 to the collection edge 30 to attach a new computer 32.

Self Organizing a Hierarchy

15 The collection of computers may be searched more rapidly if it maintains itself as a symmetrical compact shape such as that illustrated in Fig 8A rather than a collection of long thin strings as illustrated in Fig 8B. For example, a message can be broadcast to all 25 members of the collection of computers illustrated in Fig 8A within three hops from the logical middle 10 of the collection of computers. However a message requires
20 fourteen hops to be fully broadcast to the members of the collection of computers illustrated in Fig 8B.

To maintain symmetrical compactness of the collected computers, each computer in the collection follows a goal seeking behavior illustrated in Figs 9A-1 to 9H and generally defined as:

5 (a) Orbit counter clockwise at the same level of concentric polygon until another computer is encountered;

(b) Decay inward until another computer is encountered.

10 The result of the above two rules applied by each computer in the collection will result in the creation of a logical network in the shape of a near perfect quadrilateral (or other polygon, depending on the configuration).

Self Organizing States

15 In the preferred embodiment, each unit follows the goal seeking behavior illustrated in Figs 9A-1 to 9H through the logic of a state machine. A state machine is a logical mechanism that defines the existence of a plurality of physical or logical states of being and the
20 occurrences necessary to change the mechanism from one state of being to another.

For purposes of clarity, neighbor0 40 may be referred to as N0. Similarly neighbor1 41 may be referred to as N1, neighbor2 42 may be referred to as N2, and neighbor3 43 as N3. In some states, both the existence of a
5 neighbor and existence of a neighbor's neighbor determine the state. In such instances, for clarity, the neighbor0 of neighbor0 is referred to as N00. Similarly the neighbor2 of neighbor1 is referred to as N12.

Fig 9A-1 to 9A-4 illustrate the state diagram of the
10 goal seeking behavior of an individual computer relative to the collection of computers of which it is a part. In particular, Fig. 9A-1 to 9A-4 exemplify the possible state transitions at least for a quadrilateral configuration. These states, however, relate only to the logical
15 organization of the computers with respect to the logical middle 10 and not to the identify or location, at any particular time, of the top computer 11. For clarity,

Each computer in the collection of computers may exist in one of the following states based on the
20 existence and state of its neighbor units and its proximity to a radial 26:

State 1 (illustrated in Fig 9B) - The computer is searching for a found computer 28 in the collection of computers. Three strategies are used:

(a) Search a pre existing list of network addresses
5 of computers known to belong to the collection of computers.

(b) Search a cache of network addresses of computer with which the searching unit communicated when previously connected to the collection of computers.

10 (c) Use a brute force search of all possible network addresses of computers. In an embodiment using Internet addresses, the search will be made by generating random 32-bit numbers corresponding to Internet Protocol (IP) addresses and excluding the ranges of non routable IP
15 addresses.

The searching computer confirms location of a computer in the collection of computers by executing a handshake protocol consisting of the following exchange:

(a) Searching Unit - "Are you connected to the
20 collection of computers?"

(b) Queried Unit - "Yes I am, no I am not, or I don't understand."

When the searching computer locates the collection of computers, its state becomes State 2.

State 2 (illustrated in Fig 9C) - The computer has located the collection of computers and is attempting to locate the collection edge 30. The computer follows the neighbor2 42 neighbor links of the found computer 28 to locate the next closest computer to the collection edge 30. The computer will continue to follow the neighbor2 42 neighbor links of successive neighbors until a neighbor is found with no neighbor2 42 link. A neighbor without a neighbor2 42 link resides on the collection edge 30.

State 3 (illustrated in Fig 9D) - The computer is located on the collection edge 30 and on a radial 26. Since the computer is located on the collection edge 30, the computer's neighbor2 42 does not exist. Furthermore, as illustrated in Fig 9D the neighbor01 position is vacant.

At the first opportunity, the computer will move into the neighbor01 position thereby moving closer to the logical middle 10 of the collection of computers.

State 4 (illustrated in Fig 9E) - The computer is located on the collection edge 30 and therefore the computer's neighbor2 42 does not exist. The computer is

not located on the radial, but is touching it. The computer's neighbor1 41 does not exist.

At the first opportunity the computer will move into the neighbor01 position thereby moving closer to the
5 logical middle 10 of the collection of computers.

State 5 (illustrated in Fig 9F) - The computer is located on an indirect radial 27. Neighbor1 41 does not exist. Neighbor2 42 does not exist, and the neighbor01 position is filled.

10 At the first opportunity the computer will move onto the indirect radial 27 behind the neighbor01 position thereby moving closer to the logical middle 10 of the collection of computers.

State 6 (illustrated in Fig 9G) - The computer is
15 located on an indirect radial 27. Neighbor1 41 does not exist. Neighbor2 42 does not exist. The neighbor01 position is vacant.

At the first opportunity the computer will move into the neighbor01 position.

20 State 7 (illustrated in Fig 9H) - The computer is located on an indirect radial 27, but its Neighbor1 41 is a radial 26. Neighbor2 42 does not exist.

State 8 (illustrated in Fig. 9I) - The computer is located on an indirect radial 27. The N12 position is
5 vacant.

State 9 (illustrated in Fig. 9J) - The computer is located on an indirect radial 27. Neighbor1 41 exists and Neighbor1 41 is on a radial 26.

State 10 (illustrated in Fig. 9K) - The computer is located on a radial 26. Either Neighbor1 41 or Neighbor2 42 exist preventing movement.

State 11 (illustrated in Fig. 9L) - The computer is located on an indirect radial 27. Either Neighbor2 42 or N12 exist preventing movement. One neighbor must exist to prevent movement.

State 12 (illustrated in Fig. 9M) - The computer is located on a radial 26. Neighbor0 40 is vacant. Either neighbor1 41 or neighbor3 43 exist so there is a way to link to the neighbors of the missing neighbor0 40.

5 At the first opportunity, the computer will move into the vacant neighbor0 40 position.

State 13 (illustrated in Fig 9N) - The computer is located on an indirect radial 27 but is touching a radial 26. Neighbor0 40 is vacant. Either neighbor1 41 or
10 neighbor3 43 exist so there is a way to link to the neighbors of the missing neighbor0 40.

At the first opportunity, the computer will move into the vacant neighbor0 40 position.

State 14 (illustrated in Fig 9O) - The computer is
15 located on an indirect radial 27. Neighbor0 40 is vacant. Either neighbor1 41 or neighbor3 43 exist so there is a way to link to the neighbors of the missing neighbor0 40.

At the first opportunity, the computer will move into the vacant neighbor0 40 position.

20 State 15 (illustrated in fig 9P) - The computer is located on an indirect radial 27. Neighbor0 40 exists. Neighbor1 41 is vacant. Neighbor2 42 is vacant. Neighbor 01 is vacant.

At the first opportunity, the computer will move into the vacant neighbor01 position.

Bandwidth Availability Sorting

After entering one of the stable states, each
5 computer will compare its available network bandwidth with that of its Neighbor0 40. If the computer's available network bandwidth is greater than that of its Neighbor0 40, the computer and its Neighbor0 40 will swap places in the collection of computers.

10 The effect of the computers swapping places is that the collection of computers becomes sorted by available network bandwidth. The computers with the greatest amount of available network bandwidth will be closest to the logical middle 10 of the collection of computers.

15 Failure Recovery

The collection of computers can rebuild the connections necessary for broadcast and searching in the event a computer in the collection becomes disabled by a hacking attack or ceases functioning due to technical or
20 network problems.

Fig 10A illustrates the collection of computers following the disabling of several computers. The disabled computers 29 are unable to pass along broadcast messages and must therefore be replaced. Figs 10B through 5 Fig 10C illustrate the process by which the remaining computers in the collection reorganize the remaining computers.

Once the disabled computers have been removed from the interior of the collection of computers, the 10 reorganization will continue to rebuild the symmetrical compactness illustrated in Figs 9B to 9K.

Computer Locking

Each computer in the collection of computers functions as an autonomous unit asynchronously related to 15 its neighbors. In order to synchronize the movements of computers to different positions in the collection, it is necessary to prevent race conditions or ambiguous situations which could occur should a computer make a decision based on the states of its neighbors and move its 20 position at the same time the neighbor's states are changing.

For this reason a computer in the collection will
lock its neighbors from making any moves until it:

(a) Senses the state of its neighbors.

(b) Decides if a move is necessary.

5 (c) Performs the move.

The lock has a timeout to prevent a never ending lock
should the locking computer be disabled in some way before
releasing the lock.

Limiting Broadcast or Searches

10 In some cases it may be desirable to limit message
broadcast or content searches to a subset of the
collection of computers. In that case, the broadcast or
search may be limited by specifying the number of
concentric rings to search relative to the computer
15 requesting the broadcast or search.

Bandwidth Throttling

If all computers in the collection of computers do
comprehensive message broadcasts or content searches, the
communications means logically connecting the computers
20 may become congested.

Any computer in the collection of computers can view the sum total of the communications traffic between all computers. Should the communications traffic begin to approach to capacity of the logical communications channel
5 a means of reducing the traffic must be considered.

The following technique is fair to each user of the collection of computers, rewards efficient use, and penalizes overused. Before each broadcast or search is initiated, each and every computer in the collection of
10 computers will perform the following before performing a broadcast or search:

(a) Measure the logical communication channel bandwidth utilization.

(b) If the utilization is below a predetermined
15 desired threshold, proceed with the broadcast or search, otherwise,

(c) Calculate the time since the last search.

(d) If the time since the last search is less than a predetermined amount, insert a predetermined delay in
20 front of the request for broadcast or search which is related to the amount the communications channel bandwidth utilization exceed the predetermined desired threshold.

The above algorithm will assure that broadcast and search performance will degrade gracefully under heavy use of the collection of computers.

Caching Computers

5 If the collections of computers large enough and search demands frequent enough to fill the logical communication channel bandwidth, a more efficient use of bandwidth is possible by defining a subset of the collection of computers to perform a caching function.
10 The computers chosen to perform the caching function are those with faster network connections.

Fig. 11 illustrates the preferred embodiment of the caching organization. In Fig. 11 the caching computers 33 are organized into a collection of computers similar to the collection of non-caching computers shown in Fig. 2.
15 Each caching computer 33 is the logical middle of a collection of computers. As such, the caching computers 33 and the underlying collections of computers represent a hierarchy within a hierarchy or, in an alternative way of looking at it, a simple type of three dimensional
20 configuration. Each caching computer 33 caches the content index and name or other identifying characteristic

of each computer in the collection of computers which surround the caching computer 33.

Each caching computer 33 is also a member of the collection of caching computers. The collection of
5 caching computers self organizes itself so that the computer with the most available network bandwidth resides at the logical middle 10 of the collection of caching computers.

Each caching computer in the collection of caching
10 computers compares its available network bandwidth with the network bandwidth of its neighbor nearer the logical middle of the collection of caching computers. If the caching computer possesses more available network
bandwidth than its neighbor, the computer and its neighbor
15 swap positions in the collection of caching computers.

Each caching computer in the collection continuously performs the compare and swap operation, such that the collection of caching computer is always sorted with the computer having the most available network bandwidth
20 located at the logical middle of the collection of caching computer and each concentric polygon contains computers with successively less available network bandwidth.

Adding a Caching Computer

A caching computer is added to the collection of caching computers in much the same way as a computer is added to the collection of computers as described in Self
5 Organizing a Hierarchy and illustrated in Figs. 9A-1 to 9P.

A new caching computer can find the collection of caching computers using several techniques including but not limited to:

10 (a) Trying to attach to a known list of available caching computers in the collection of caching computers.

(b) Trying to attach to a caching computer from a list of caching computers with which this computer has previously communicated.

15 (c) Seeking a computer connected to the collection of caching computers by broadcasting queries to network addresses created from an algorithm such as a random number generator.

When a new caching computer locates a computer
20 belonging to the collection of caching computers the found caching computer will reside either on a radial 26 or an indirect radial 27. The tail 41 of the found caching computer 28 will point toward the collection edge 30. By

following the found caching computer's 28 neighbor's tail to head one after another, the collection edge 30 may be located. The new caching computer is attached to the collection of caching computers at the edge 30.

5 To maintain symmetrical compactness of the collected computers, each computer in the collection follows a goal seeking behavior as illustrated in Fig 9:

(a) Orbit counter clockwise at the same level of concentric polygon until another computer is encountered;

10 (b) Decay inward until another computer is encountered.

A newly added caching computer will have a large amount of available network bandwidth, and as such will swap positions with its neighbors until it reaches the logical middle of the collection of caching computers. Preferably, a newly added computer will be designated as a caching computer based on whether the newly added computer possesses an available amount of bandwidth greater than some threshold. Newly added computers that do not possess an amount of bandwidth over the threshold are added as non-caching computers.

15
20

Adding a Non-caching Computer

In a caching system, a new computer 32 locates the collection of computers and then follows the head to toe links until it reaches the logical middle 10 of its collection of computers. The logical middle will be a
5 caching computer that also belongs to the collection of caching computers.

The new computer 32 will follow the head to toe links of the collection of caching computers until it reaches the logical middle of the collection of caching computers.

10 The new computer 32 will provide the caching computer with its network address, the index of its content, and its name or other identifying characteristics. The new computer 32 will then follow a radial 26 of the collection of non-caching computers attached to the logical middle of
15 the collection of caching computers. The new computer 32 will attach itself to the edge 30 of the collection of computers and begin the self-organizing logic to seek a position.

Self Balancing Cache

20 Due to the bandwidth availability comparison logic, the computer at the logical middle of the collected caching computers will always be the caching computer with

the most available bandwidth. New non-caching computers will always be added to the collection of non-caching computers attached to this caching computer.

As more non-caching computers are added to the
5 caching computer with the most available bandwidth, it will eventually have less available bandwidth than another caching computer in the collection of caching computers and be replaced at the middle of the collection of caching computers by a computer with more bandwidth available.

10 In the above fashion, the collection of caching computers will always maintain an optimally balanced distribution of non-cached computers attached to each caching computer.

Unavailable Computer

15 From time to time, a caching computer will poll each of the computers in its collection of non-caching computers. When a computer that is a member of a collection of computers headed by a caching computer becomes unavailable or unusable through some sort of
20 failure, the caching computer will remove the unavailable computers cached content index, name, and other identifying characteristics.

Unavailable Caching Computer

From time to time, a caching computer will fail to respond to communications from non-caching computers in its collection of computers. The caching computer will be
5 replaced by one of the non-caching computers in its collection of computers using the algorithms illustrated in Fig 9A-1 to 9P.

Should there be no more non-caching computers available in the disabled caching computers collection of
10 computers, the collection of caching computers will fill in the void using the logic illustrated in Fig 9A-1 to 9P.

Auto-update

The invention embodies a feature by which a computer compares its version of software that controls the
15 collection of computers to that of the neighbor 40,41,42,43 with which it is preparing to communicate. If the computer requesting the communication possesses a more recent version of the software that controls the collection of computers, then that version of software is
20 transferred to the neighbor 40,41,42,43.

If the neighbor 40,41,42,43 to which communications is being requested possesses a more recent version of the

software that controls the collection of computers, then the more recent version of the software is transferred from the neighbor 40,41,42,43 to the computer requesting the communications.

5 In this fashion, a software update applied to any computer in the collection of computers will propagate through the collection of computers at about the same rate as a message broadcast.

10 Distributing Executables Across the Collection of Computers

 It is possible using the auto-update mechanism described above to distribute executable programs to all computers in the collection of computers.

15 It is possible using the auto-update mechanism described above to distribute executable programs to a subset of all the computers in the collection of computers by limiting the distribution to computers within a fixed number of concentric polygons of the computer initiating the distribution.

20 The computers receiving executable programs using the auto-update mechanism may be commanded to perform certain operations using the executable programs distributed using

the auto-update mechanism including executing the programs to perform a calculation or other operation and returning a result to the computer initiating the distribution of the executable.

5 Broadcast of Streaming Media

One embodiment of the invention provides for the one to many broadcast of streaming media. Streaming media include but are not limited to WEB radio and corporate WEBcasts.

10 Current broadcast technology requires the originator posses a large server and high speed network connection necessary to open the required channel to each and every destination as illustrated in Fig. 12. Therefore a doubling of destinations requires a doubling of the
15 originator's network bandwidth.

The invention offers an efficient means to broadcast streaming media by sharing the necessary bandwidth and computing power with each destination computer. Fig. 13 illustrates the streaming media possible with the
20 collection of computers created by the invention.

The advantages of the invention over existing solutions are:

(a) The originator's network bandwidth need only be as large as that necessary for a single stream of the media.

(b) The number of destinations may be increased
5 irrespective of the bandwidth of the originator's network connection.

(c) The number of destinations may be increased virtually without limit.

Making a Streaming Media Connection

10 A WEB radio station is an existing example of streaming media. With current technology, a user with a media player program contacts the station's WEB page, provides the user's network address, and requests to be connected to the station's streaming media server. The
15 media server opens a connection and begins sending the media information to the user's media player at the user's network address. The media information is virtually the same as that sent to every other user.

When using one embodiment of the invention, a user
20 with a media player program and a program implementing the invention contacts the station's WEB page, provides the user's network address, and requests to be connected to

the station's streaming media server. The station streaming media server is using a version of the invention.

5 The media server directs the new user to the logical middle of the collection of computers 10. The new user's computer 32 follows the protocol described in Figs. 9A-1 to 9P to attach to the collection of computers. Upon attachment to the collection of computers, the new computer 32 will begin receiving the streaming media from
10 the neighbor to which its head 40 is attached.

When yet another computer is attached to the new computer 32, the new computer 32 will forward the streaming media to the added computer.

15 When a Computer Disconnects or Becomes Unavailable

When a computer in the collection of computers used to broadcast streaming media disconnects or otherwise becomes unavailable, the invention replaces the unavailable computer in the collection of computers using
20 the protocol illustrated in Figs. 10B and 10C.

Bandwidth Requirements

Each computer in the collection of computers requires an available outbound network bandwidth the same as the inbound network bandwidth of the streaming media. The exceptions are computers which reside on radials 26 which
5 require twice the available outbound bandwidth as inbound bandwidth.

Advantages of the Invention

The invention possesses the following advantages over the previous art include:

- 10 (a) The ability to broadcast a message from any computer in a collection of computers to broadcast to the balance of the collection by distributing the computing power and communications bandwidth required by the broadcast to the collection of computers.
- 15 (b) The ability of any computer in a collection of computers to search for content in a collection of computer by distributing the computing power and communication bandwidth required by the search to the collection of computers.
- 20 (c) The ability of any computer in a collection of computers to search for another computer in the collection of computers by a name or other identifying characteristic

by distributing the computing power and communication bandwidth required by the search to the collection of computers.

5 (d) The ability to self organize computers into a collection of computers such that the computing power and communication bandwidth of the collection may be used to broadcast from one computer to the balance of computers in the collection; such that the computing power and communication bandwidth of the collection may be used to
10 search for content existing on the collection of computers; and such that the computing power and communication bandwidth of the collection may be used to locate a computer in the collection by name or other identifying characteristic.

15 (e) The ability to self organize computers which possess transient network addresses into a collection of computers such that the computing power and communication bandwidth of the collection may be used to broadcast from one computer to the balance of computers in the
20 collection; such that the computing power and communication bandwidth of the collection may be used to search for content existing on the collection of computers; and such that the computing power and

communication bandwidth of the collection may be used to locate a computer in the collection by name or other identifying characteristic.

(f) The ability of an organized collection of
5 computers to self repair itself after some computers become inoperative to an attack or technical malfunction. Referring now to FIGURE 1,

Although a preferred embodiment of the method and apparatus of the present invention has been illustrated in
10 the accompanying Drawings and described in the foregoing Detailed Description, it is understood that the invention is not limited to the embodiment disclosed, but is capable of numerous rearrangements, modifications, and substitutions without departing from the spirit of the
15 invention as set forth and defined by the following claims.